PathAddBackslash

Return value buffer must be large enough to store returned path

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4239 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• Buffer Overflow • Unconditional
Software Context	• File Path Management
Location	• shlwapi.h
Description	When using the PathAddBackslash() function, the in/out buffer used to return the path must be large enough to hold the returned value. PathAddBackslash() and variants add characters (backslash) in place to the path name that is passed in. The lpszPath parameter must be at least MAX_PATH *characters* (not bytes) in length to ensure it is large enough to hold the returned string. The behavior of PathAddBackslash() when the supplied string is already MAX_PATH long is undefined. It is not clear if the function checks the length of the existing data or not. This is potentially a vulnerability in some implementations.
APIs	**Function Name** — **Comments** PathAddBackslash PathAddBackslashA PathAddBackslashW ATLPath::AddBackslash — Overloaded wrapper to PathAddBackslash
Method of Attack	Attacker can cause a buffer overflow if the path variable is not long enough to hold the variable. It's not clear whether the adding of a backslash can be leveraged into an exploit, however. Undefined behavior provides a target for part of an attack.
Exception Criteria	

[1] http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

<table>
<tr><td><b>Solutions</b></td><td><table>
<tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr>
<tr><td>Whenever using PathAddBackslash() or variants.</td><td>Ensure that buffer is declared as at least MAX_PATH characters in size.</td><td>Effective.</td></tr>
</table></td></tr>
<tr><td><b>Signature Details</b></td><td>LPTSTR PathAddBackslash(
LPTSTR lpszPath
);
inline char* AddBackslash(
char* pszPath
);
inline wchar_t* AddBackslash(
wchar_t* pszPath
);</td></tr>
<tr><td><b>Examples of Incorrect Code</b></td><td><code>TCHAR buffer_1[] = TEXT("C:\\dir_name\\dir_name\\file_name"); // Buffer is too small</code>
<code>LPTSTR lpStr1;</code>
<code>lpStr1 = buffer_1;</code>
<code>PathAddBackslash(lpStr1);</code></td></tr>
<tr><td><b>Examples of Corrected Code</b></td><td><code>TCHAR buffer_1[MAX_PATH] = TEXT("C:\\dir_name\\dir_name\\file_name"); // Buffer is safely sized</code>
<code>LPTSTR lpStr1;</code>
<code>lpStr1 = buffer_1;</code>
<code>PathAddBackslash(lpStr1);</code></td></tr>
<tr><td><b>Source Reference</b></td><td>• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/shlwapi/path/pathaddbackslash.asp[2]</td></tr>
<tr><td><b>Recommended Resource</b></td><td>• MSDN reference for ATLPath::AddBackslash[3]</td></tr>
<tr><td><b>Discriminant Set</b></td><td><table>
<tr><td><b>Operating System</b></td><td>• Windows</td></tr>
<tr><td><b>Languages</b></td><td>• C
• C++</td></tr>
</table></td></tr>
</table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

[1] mailto:copyright@cigital.com